Adaptively hybrid fractal image coding

Qiang Wang,[*] **Guohua Jin, and Sheng Bi**

College of Information Science and Technology, Dalian Maritime University, Dalian, 116026, China

Abstract. In this study, an adaptively hybrid method was proposed to improve the performance of fractal coding methods. First, we found that the range blocks with large variances (RBLVs) play a crucial role in degrading decoded images, and the effect of the remaining range blocks with small variances (RBSVs) can be ignored. Second, RBLVs were designed to be encoded in an extended domain block pool (EDBP), and the remaining RBSVs were encoded with the no-search fractal encoding method. Moreover, an effective method to adaptively divide the range blocks into the above two categories was proposed. Finally, four fractal coding methods were adopted to assess the performance of the proposed method. Experimental results show that, compared with the previous methods, the proposed method can achieve better-decoded image quality with fewer bits per pixel and fewer computations.

Keywords: Fractal image coding; Collage error; No-search; Decoded image quality

*E-mail: wangqiang2011@dlmu.edu.cn

1 Introduction

Different from conventional image compression techniques, fractal image coding is characterized by its novel idea, potential high compression ratio, fast decoding, and resolution independence.[1-4] Thus, it has attracted much attention from researchers globally. However, owing to its high computational complexity in the encoding process, fast fractal image coding becomes one of the most important topics in this field. In recent years, many researchers have attempted to overcome this problem by converting exhaustive searching into local matching.[5-7] In this case, as the number of block-matching operations decreases significantly, the encoding process can be done in a shorter time. To further accelerate the encoding process, no-search fractal image encoding should not perform the block matching operations, and a domain block is directly appointed as the best-matched domain block. Thus, no-search fractal image encoding can achieve real-time encoding and a higher compression ratio at the expense of the quality of the decoded images.[8] In addition to image compression, fractal coding methods have also been gradually implemented in other image processing applications such as image denoising,[9-15] image retrieval,[16-18] image hashing,[19,20] watermarking,[21-23] image super resolution[24-28], and MRI images.[29]

In this study, we found that the range blocks with large variances (RBLVs) play a crucial role in determining the quality of decoded images. However, the effect of the remaining range blocks with small variances (RBSVs) can be ignored. Therefore, we designed an extended domain block pool (EDBP) for RBLVs. Because EDBP can effectively reduce collage errors, the quality of decoded images can be expected to improve. Furthermore, to accelerate the encoding process and reduce the bits per pixel, we adopted the no-search fractal coding method to encode RBSVs. Moreover, the bottom limit of the actual percentage of an accumulated collage error (BLAPACE) was proposed to adaptively divide the range blocks into the above two categories. Finally, four fractal encoding methods, Jacquin's,[1] Chaurasia's,[5] Zheng's[6], and Gupta's methods,[7] were adopted to assess the effectiveness of the proposed method. Experimental results show that compared with the previous methods, the proposed method can achieve a better quality of decoded images with fewer bits per pixel and fewer computations.

The remainder of the study is organized as follows: an introduction about conventional fractal image coding is reviewed in Sec.2. In Sec.3, a hybrid fractal encoding method is proposed. By analyzing the importance of variances, RBLVs are encoded in EDBP, and

RBSVs are encoded with no-search fractal image coding. An adaptive method is also proposed to divide the range blocks into the above two categories. In Sec.4, the experimental procedures and performance of the proposed method are described and discussed in detail. Finally, the conclusion is given in Sec.5.

2 Conventional fractal image coding

Fractal image coding aims to establish an iterated function system (IFS) whose fixed point can well approximate the input image. It mainly consists of a series of block-matching operations between range and domain blocks. First, the range blocks divide the $M \times N$ input image uniformly into nonoverlapping $B \times B$ blocks; $\boldsymbol{R}_i$, i=1,2,3,..., NumR, and domain blocks; $\boldsymbol{D}_j$, j=1,2,3,..., NumD, can be obtained by sliding a $2B \times 2B$ window over the input image from left to right and top to bottom, where NumR and NumD represent the total numbers of range and domain blocks, respectively. In this study, the sliding step δ was set as $2B$. To make domain blocks match range blocks well, domain blocks are first contracted to the same size as range blocks and then extended with eight isometric transformations, which consist of the identity, rotations through 90°, 180°, and 270°, and reflections about the vertical axis, horizontal axis, first diagonal, and second diagonal. Figure 1 illustrates the eight transformations of an image. After performing affine transformations on the contracted domain blocks for arbitrary range blocks, the corresponding best-matched domain block can be obtained by minimizing the following function

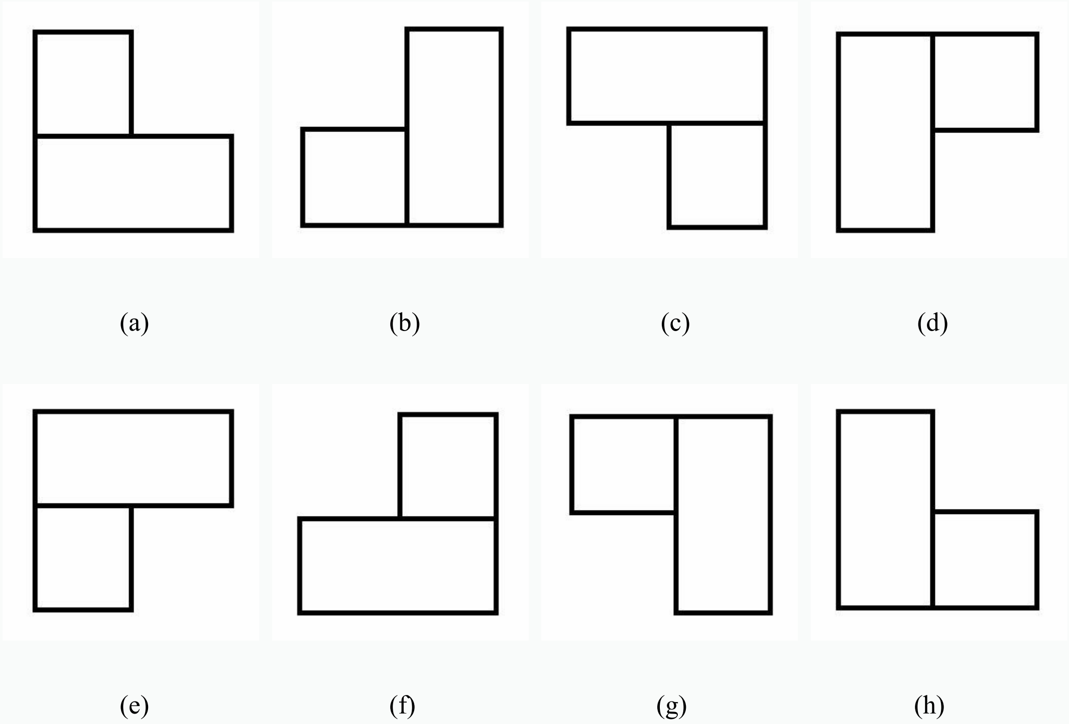
$$\mathrm{CE}(\boldsymbol{R}) = \min_{\boldsymbol{D}} \left[\min_{s,o} \left(\frac{1}{B^2} \| \boldsymbol{R} - s\boldsymbol{D} - o\boldsymbol{I} \|^2 \right) \right] \quad 1$$

where $\boldsymbol{I}$ denotes a $B \times B$ matrix whose elements are all ones. s and o denote the scaling and offset coefficients for the affine transformation, respectively. CE(***R***) is the collage error of the range block ***R***, and ***D*** denotes the domain block in the conventional domain block pool (CDBP). With the least square method, we can calculate *s* and *o* as

$$s=\left\langle \boldsymbol{R}-\bar{r}\boldsymbol{I},\quad \boldsymbol{D}-\bar{d}\boldsymbol{I}\right\rangle \Big/ \left\|\boldsymbol{D}-\bar{d}\boldsymbol{I}\right\|^{2},\quad o=\bar{r}-s\bar{d} \qquad 2$$

where $\langle \bullet,\bullet \rangle$ denotes the inner product. $\bar{r}$ and $\bar{d}$ are the mean values of ***R*** and ***D***, respectively. Figure 2 illustrates the above fractal encoding process.

At the decoding phase, an arbitrary image with the same size as the input image can be selected as the initial image. The same transformations from the best-matched domain to the corresponding range blocks as those in the encoding process were applied recursively to the initial image. After approximately 10 iterations, the decoding process converged into the final decoded image. As an example, we used 256 × 256 Peppers as the input image. If the rotated 256 × 256 Peppers were chosen as the initial image, as shown in Fig.3(a), the first five iteration images obtained in the decoding process are shown in Fig.3(b)-(f), respectively.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig.1. Eight isometric transformations. (a)-(h) represents the identity, rotations through 90° 180°, 270°, and reflections about the horizontal axis, vertical axis, first diagonal, and second diagonal, respectively.

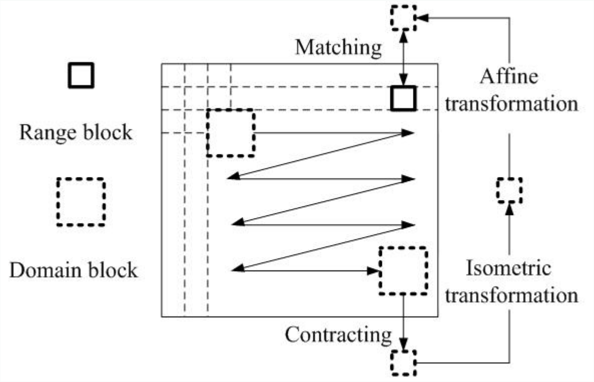
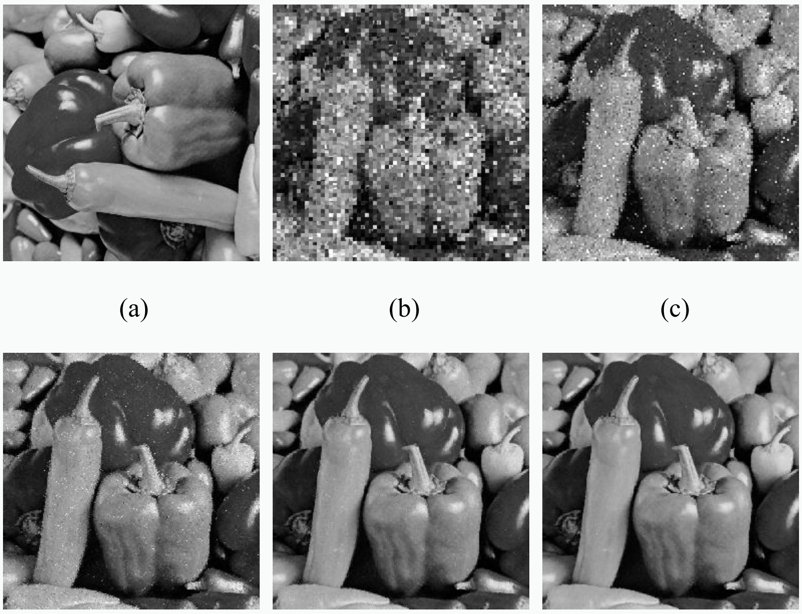


Fig.2. Illustration of the fractal encoding process.



(d) (e) (f)

Fig.3. Fractal decoding process. (a) Initial image. (b)-(f) First five iteration images in the decoding process.

3 Proposed method

The peak signal-to-noise ratio (PSNR) was adopted in this study to measure the quality of decoded images as follows:

$$\mathrm{PSNR} = \log_{10}\left(255^2 \Big/ \frac{1}{M \times N}\sum_{j=1}^{N}\sum_{i=1}^{M}\left(\boldsymbol{f}^{\mathrm{Original}}(i,j) - \boldsymbol{f}^{\mathrm{Decoded}}(i,j)\right)^2\right) \quad 3$$

where $\boldsymbol{f}^{\text{Original}}$ and $\boldsymbol{f}^{\text{Decoded}}$ denote the original and decoded images, respectively, and M and N are the height and width of the images, respectively. Moreover, we defined the average collage error (ACER) as the average of the collage errors of all range blocks, which can be formulated as

$$\text{ACER} = \sum_{i=1}^{\text{NumR}} \text{CE}(\boldsymbol{R}_i) \Big/ \text{NumR} \qquad 4$$

where NumR denotes the total number of range blocks. Furthermore, a logarithmic relationship between ACER and PSNR of the decoded image was established as[30]

$$\text{PSNR} = \beta_1 + \beta_2 \log_{10}\left(\text{ACER}\right) \qquad 5$$

where β_1 and β_2 are constant coefficients. Because β_2 is a negative value, we know that a smaller ACER implies better-decoded image quality. Eqs.[4, 5] imply that the PSNR of the decoded image can be directly determined by the sum of all collage errors.

For arbitrary $\boldsymbol{R}$, by substituting [2] into [1], we have

$$\begin{aligned}
\text{CE}(\boldsymbol{R}) &= \frac{1}{B^2}\left\|\boldsymbol{R} - s\boldsymbol{D} - o\boldsymbol{I}\right\|^2 \\
&= \frac{1}{B^2}\left(\left\|\boldsymbol{R} - \bar{r}\boldsymbol{I}\right\|^2 - s^2\left\|\boldsymbol{D} - \bar{d}\boldsymbol{I}\right\|^2\right) \\
&= \frac{1}{B^2}\left\|\boldsymbol{R} - \bar{r}\boldsymbol{I}\right\|^2\left(1 - \frac{\left|\left\langle \boldsymbol{R} - \bar{r}\boldsymbol{I}, \boldsymbol{D} - \bar{d}\boldsymbol{I}\right\rangle\right|^2}{\left\|\boldsymbol{R} - \bar{r}\boldsymbol{I}\right\|^2\left\|\boldsymbol{D} - \bar{d}\boldsymbol{I}\right\|^2}\right) \\
&= \frac{1}{B^2}\left\|\boldsymbol{R} - \bar{r}\boldsymbol{I}\right\|^2\left(1 - \text{LCC}^2\right) \le \frac{1}{B^2}\left\|\boldsymbol{R} - \bar{r}\boldsymbol{I}\right\|^2
\end{aligned} \qquad 6$$

where LCC denotes the linear correlation coefficient between $\boldsymbol{R}$ and $\boldsymbol{D}$. From Eq.[6], we can see that for an arbitrary $\boldsymbol{R}$, its variance provides a top limit for its collage error, and the range blocks with large collage errors should have correspondingly large variances. Thus, only RBLVs can lead to large collage errors. If we use LCC_{Min} to denote the minimum of the

LCCs of all range blocks, we obtain the maximum of CE($\boldsymbol{R}$) as

$$\mathrm{CE}\left(\boldsymbol{R}\right)_{\mathrm{Max}}=\frac{1}{B^2}\left\|\boldsymbol{R}-\bar{r}\boldsymbol{I}\right\|^2\left(1-\left(\mathrm{LCC}_{\mathrm{Min}}\right)^2\right) \tag{7}$$

Combining Eqs.[6, 7], we have the following inequality

$$\mathrm{CE}(\boldsymbol{R})\leq\mathrm{CE}(\boldsymbol{R})_{\mathrm{Max}} \tag{8}$$

In the encoding process, we sorted the range blocks by their variances from largest to smallest and encoded each range block separately. Subsequently, we obtained the accumulated collage error (ACE) as

$$\mathrm{ACE}=\sum_{i=1}^{\mathrm{Num1}}\mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Coded}}) \tag{9}$$

where Num1 denotes the number of coded range blocks. From Eqs.[8, 9], we have

$$\mathrm{ACE}\leq\mathrm{ACE}_{\mathrm{Max}} \tag{10}$$

where

$$\mathrm{ACE}_{\mathrm{Max}}=\sum_{i=1}^{\mathrm{Num1}}\mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Coded}})_{\mathrm{Max}}=\sum_{i=1}^{\mathrm{Num1}}\left\|\boldsymbol{R}_i^{\mathrm{Coded}}-\bar{r}_i^{\mathrm{Coded}}\boldsymbol{I}\right\|^2\left(1-\left(\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}}\right)^2\right) \tag{11}$$

where $\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}}$ denotes the minimum of the LCCs of the coded range blocks. Similarly, for the remaining uncoded range blocks, the corresponding ACE can be defined as

$$\mathrm{ACE}^{\mathrm{Uncoded}}=\sum_{i=1}^{\mathrm{Num2}}\mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}}) \tag{12}$$

where Num2 denotes the number of uncoded range blocks. Similar to Eq.[10], we obtain

$$\mathrm{ACE}^{\mathrm{Uncoded}}\leq\mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Uncoded}} \tag{13}$$

where

$$\mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Uncoded}}=\sum_{i=1}^{\mathrm{Num2}}\mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}})_{\mathrm{Max}}=\sum_{i=1}^{\mathrm{Num2}}\left\|\boldsymbol{R}_i^{\mathrm{Uncoded}}-\bar{r}_i^{\mathrm{Uncoded}}\boldsymbol{I}\right\|^2\left(1-\left(\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Uncoded}}\right)^2\right) \tag{14}$$

where $\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Uncoded}}$ denotes the minimum of the LCCs of uncoded range blocks. Moreover, we have

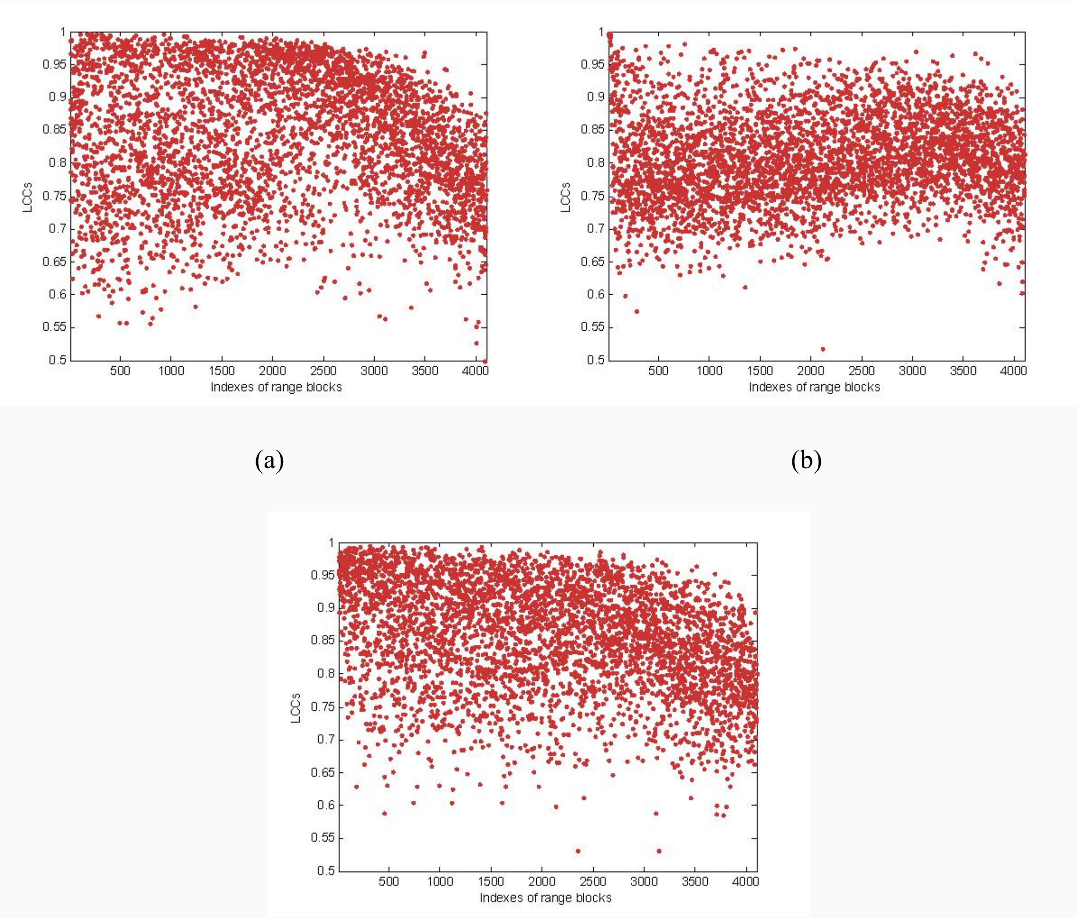
$$\mathrm{LCC}_{\mathrm{Min}} \le \mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Uncoded}} \tag{15}$$

By replacing $\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Uncoded}}$ with $\mathrm{LCC}_{\mathrm{Min}}$, Eq. (14) can be rewritten as

$$\mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Estimated\_1}} = \sum_{i=1}^{\mathrm{Num2}} \mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}})_{\mathrm{Max}} = \sum_{i=1}^{\mathrm{Num2}} \left\|\boldsymbol{R}_i^{\mathrm{Uncoded}} - \overline{r}_i^{\mathrm{Uncoded}}\boldsymbol{I}\right\|^2 \left(1-\left(\mathrm{LCC}_{\mathrm{Min}}\right)^2\right) \tag{16}$$

According to Eq. 15 and by comparing Eqs. 14, 16, we have

$$\mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Uncoded}} \le \mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Estimated\_1}} \tag{17}$$



(a)

(b)

(c)

Fig.4. Schematic of the LCCs for the three test images.

Generally, for one input image, there are a large number of range blocks. For example, if the size of range blocks is set as 4 × 4, 256 × 256 images consist of 4,096 range blocks. We selected three 256 × 256 test images, Barb, Baboon, and Man, for example, and Figures 4(a), (b), and (c) illustrate the distributions of the LCCs for the three test images, respectively. Clearly, following the encoding order, the LCCs distribute randomly. If sufficient range blocks were encoded, the minimum of the LCCs of all range blocks, $\mathrm{LCC}_{\mathrm{Min}}$, can be approximated with the minimum of the LCCs of coded range blocks, $\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}}$, and we obtain

$$\mathrm{LCC}_{\mathrm{Min}} \approx \mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}} \tag{18}$$

By replacing $\mathrm{LCC}_{\mathrm{Min}}$ with $\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}}$, Eq.[16] can be rewritten as

$$\mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Estimated\_2}} = \sum_{i=1}^{\mathrm{Num2}} \mathrm{CE}\left(\boldsymbol{R}_i^{\mathrm{Uncoded}}\right)_{\mathrm{Max}} = \sum_{i=1}^{\mathrm{Num2}} \left\| \boldsymbol{R}_i^{\mathrm{Uncoded}} - \bar{r}_i^{\mathrm{Uncoded}} \boldsymbol{I} \right\|^2 \left(1 - \left(\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}}\right)^2\right) \tag{19}$$

According to Eq.[18] and by comparing Eqs.[16, 19], we obtain

$$\mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Estimated\_1}} \approx \mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Estimated\_2}} \tag{20}$$

According to Eqs.[17, 20], we have

$$\mathrm{ACE}^{\mathrm{Uncoded}} \le \mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Estimated\_2}} \tag{21}$$

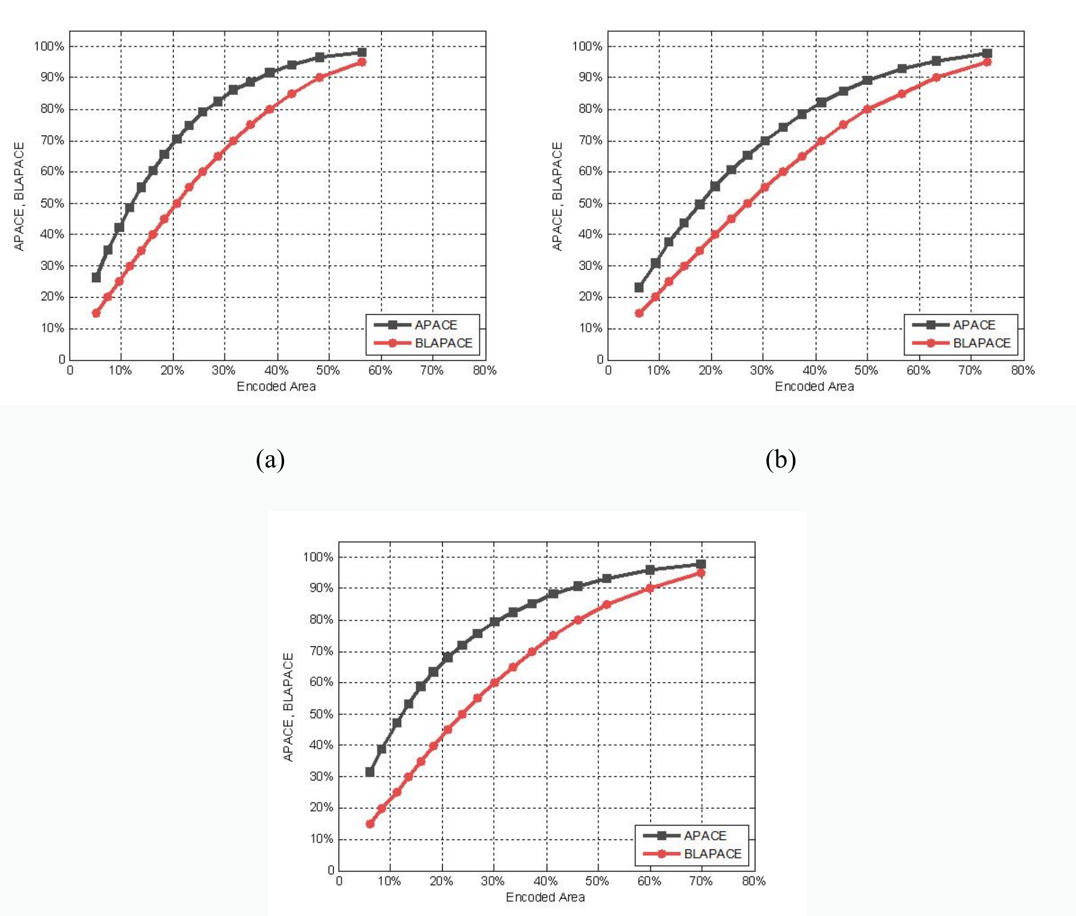
If we define the actual percentage of accumulated collage error (APACE) as

$$\mathrm{APACE} = \frac{\mathrm{ACE}}{\mathrm{ACE} + \mathrm{ACE}^{\mathrm{Uncoded}}} \tag{22}$$

By Eqs. [21, 22], we can get the BLAPACE as

$$\underset{\mathrm{BLAPACE}}{\frac{\mathrm{ACE}}{\mathrm{ACE} + \mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Estimated\_2}}}} \le \underset{\mathrm{APACE}}{\frac{\mathrm{ACE}}{\mathrm{ACE} + \mathrm{ACE}^{\mathrm{Uncoded}}}} \tag{23}$$

From Eq.[23], we know that in the encoding process, when BLAPACE reaches a large percentage, such as 95%, APACE will be more than 95%, and this implies that the coded range blocks contribute more than 95% of the total collage errors. If we can reduce the collage errors of the coded range blocks effectively, the total collage errors of all range blocks will decrease as well.



(a)

(b)

(c)

Fig.5. APACE and BLAPACE versus the percentage of coded area for the three test images, respectively.

Figure 5(a), (b), and (c) show the BLAPACE and APACE versus the percentage of encoded area for the three test images in Fig.4. We can easily observe the following key points:

1) Coded RBLVs contribute most of the collage errors. From Fig.5, we can observe that in the sorted range block sequences, the first 50% of range blocks contribute 97%, 89%, and 93% of the total collage errors for the three images. This implies that RBLVs play a more crucial role than the remaining RBSVs. Thus, we divided the range blocks into two categories:

RBLVs and RBSVs, which contribute most and a little of the collage errors, respectively. Furthermore, we adopted two different strategies to encode the above two types of range blocks: a) For RBLVs, their corresponding best-matched domain blocks were searched in the EDBP, which consisted of the CDBP and supplementary domain block pool (SDBP). Besides the CDBP as in Sec.2, we also designed an SDBP, which can be constructed with the following two steps: 1. Slide a $4B \times 4B$ window over the input image from left to right and top to bottom, and the sliding step was set to $4B$. 2. Contract all the domain blocks to the size of $B \times B$ by averaging every 4×4 pixels, and perform eight isometric transformations on them. For arbitrary RBLV, its best-matched domain block was searched in EDBP. Compared with CDBP in Sec.2, EDBP can provide more domain blocks as the candidates for the best-matched domain block, and the corresponding collage error can be expected to decrease effectively. b) For RBSVs, we adopted the no-search encoding method to encode them. As in Fig.6, the domain block with the same center as the range block was directly appointed as the best-matched domain block. From Eq. (6), we know that small variances will provide upper limits for their corresponding collage errors. This implies that small collage errors can be maintained. However, the no-search method required fewer bits per pixel and computations. Finally, for the whole image, compared with the conventional method in Sec.2, we can expect to obtain better-quality decoded images with fewer bits per pixel and fewer computations.

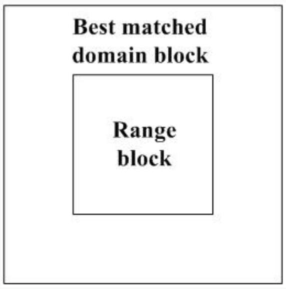


Fig.6. Range block and its associated best-matched domain block.

2) Consistent with Eq. [23], BLAPACE is often smaller than APACE. When BLAPACE reaches a large percentage, such as 95%, APACE should be larger than 95%, and this implies that the coded range blocks contribute more than 95% of the total collage errors. The range blocks whose BLAPACE reaches 95% of the total collage errors were classified as RBLVs, and the remaining ones were considered RBSVs. As shown in Fig.5, when BLAPACE reaches 95%, the coded range blocks comprise 56%, 73%, and 70% of the total number of range blocks. This implies that for different images, the numbers of RBLVs differ significantly for the three images. Thus, by Eq. (23), the range blocks can be adaptively divided into the above two categories.

4 Experiment

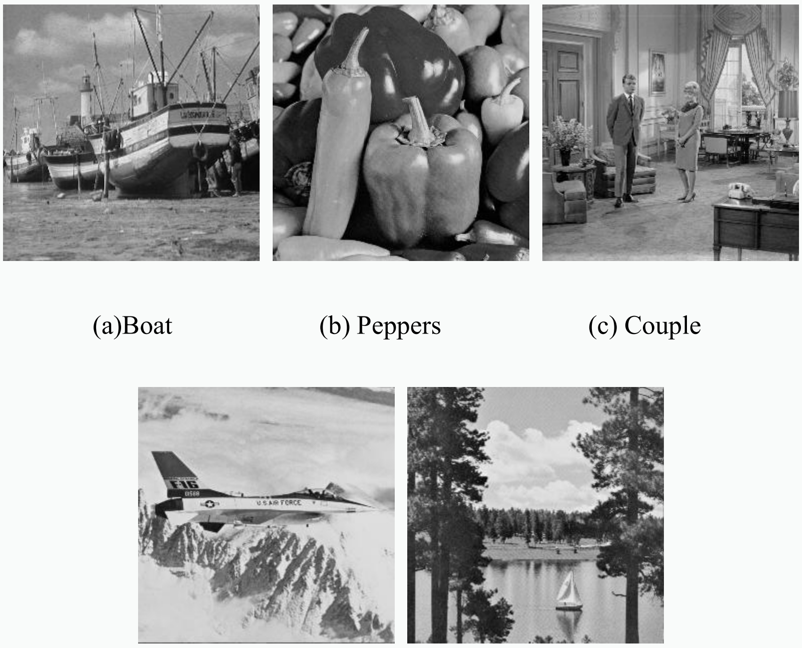
In this Section, five-256 × 256 images (Fig.7, boat, peppers, couple, airplane, and lake) were selected. The sizes of range and domain blocks were set at 4 × 4 and 8 × 8, respectively. For RBLVs, we used 3, 5, and 7 bits to quantize their isometric, scaling, and offset coefficients, respectively. Eleven bits were used to represent its position information. For RBSVs, we used 2 and 8 bits to quantize the scaling and offset coefficients, respectively. Four fractal coding methods, Jacquin's,[1] Chaurasia's,[5] Zheng's[6], and Gupta's methods,[7] were adopted to assess the performance of the proposed method by the quality of decoded images, bits per pixel, and the percentage of the required computations. The detailed procedures for the proposed method are as follows:

Step 1: For a fractal coding method, given an input image, divide it uniformly into $B \times B$

range blocks. Calculate the variances of all range blocks and sort them by their variances from largest to smallest.

Step 2: Establish CDBP and SDBP as in Sections 2 and 3, respectively, and then EDBP can be obtained by combining CDBP and SDBP.

Step 3: Encode the range blocks separately, and their corresponding best-matched domain blocks are obtained by searching within EDBP. By Eq. (23), when BLAPACE reaches 95%, the coded range blocks are classified as RBLVs, and the remaining RBSVs are encoded with a no-search fractal encoding method.



(a)Boat (b) Peppers (c) Couple

(d) Airplane (e) Lake

Fig.7 Five images used in the experiment.

Table 1 illustrates the comparison results between Jacquin’s and the proposed methods. Clearly from the second and third rows of Tab.1, the average decoded image quality for the five test images increased from 29.15 to 29.35 dB, whereas the proposed method provided better decoded image quality than Jacquin’s method. RBLVs contributed to 95% of the total collage errors and played a key role in determining the total collage errors. However, compared with Jacquin’s method, the best-matched domain blocks of RBLVs in the proposed method were found in a larger domain block pool, EDBP, and smaller collage errors were obtained. The analyses are as follows: By Eqs. (4) and (5), for all range blocks, we define the following measure to compare the proposed with Jacquin’s methods:

$$\text{Measure}=\text{ACE}^{\text{Proposed}}-\text{ACE}^{\text{Jacquin's}}=\sum_{i=1}^{\text{NumR}}\text{CE}^{\text{Proposed}}(\boldsymbol{R}_i)-\sum_{i=1}^{\text{NumR}}\text{CE}^{\text{Jacquin's}}(\boldsymbol{R}_i) \tag{24}$$

where $\text{ACE}^{\text{Proposed}}$ and $\text{ACE}^{\text{Jacquin's}}$ denote the ACEs of all range blocks for the proposed method and Jacquin’s method, respectively. If Measure is < 0, the proposed method can provide a smaller ACE and achieve a better block-matching effect than Jacquin’s method. Furthermore, for RBLVs, we define the following measure, $\text{Measure}^{\text{RBLVs}}$, to compare the proposed with Jacquin’s methods:

$$\begin{aligned}\text{Measure}^{\text{RBLVs}}&=\text{ACE}^{\text{RBLVs\_Proposed}}-\text{ACE}^{\text{RBLVs\_Jacquin's}}\\&=\sum_{i=1}^{\text{NumRBLV}}\text{CE}^{\text{RBLVs\_Proposed}}(\boldsymbol{R}_i)-\sum_{i=1}^{\text{NumRBLV}}\text{CE}^{\text{RBLVs\_Jacquin's}}(\boldsymbol{R}_i)\end{aligned} \tag{25}$$

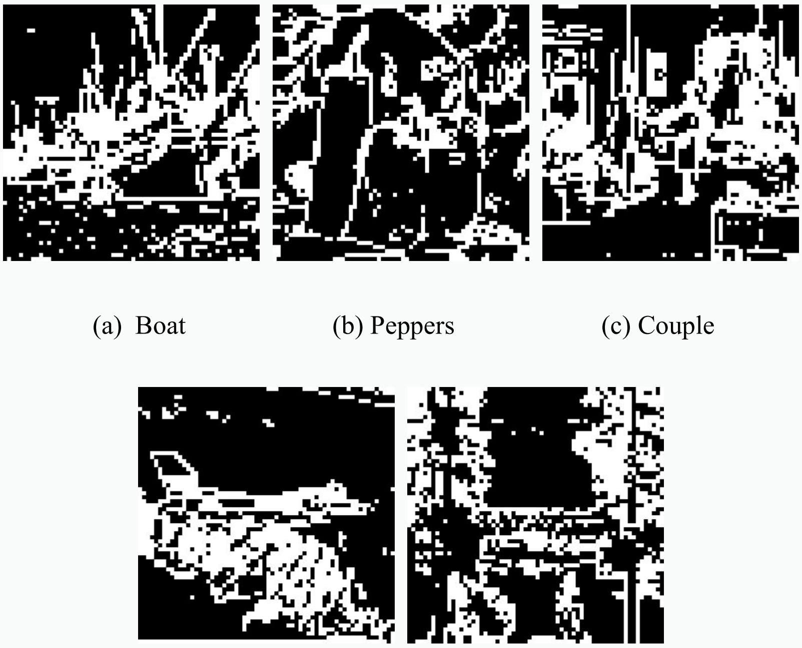
where $\text{ACE}^{\text{RBLVs\_Proposed}}$ and $\text{ACE}^{\text{RBLVs\_Jacquin's}}$ denote the ACEs of RBLVs for the proposed and Jacquin’s methods, respectively. NumRBLV denotes the number of RBLVs. Similarly, for RBSVs, we define the following measure, $\text{Measure}^{\text{RBSVs}}$, to compare the proposed with Jacquin’s methods:

$$\begin{aligned}\text{Measure}^{\text{RBSVs}}&=\text{ACE}^{\text{RBSVs\_Proposed}}-\text{ACE}^{\text{RBSVs\_Jacquin's}}\\&=\sum_{i=1}^{\text{NumRBSV}}\text{CE}^{\text{RBSVs\_Proposed}}(\boldsymbol{R}_i)-\sum_{i=1}^{\text{NumRBSV}}\text{CE}^{\text{RBSVs\_Jacquin's}}(\boldsymbol{R}_i)\end{aligned} \tag{26}$$

where $\text{ACE}^{\text{RBSVs\_Proposed}}$ and $\text{ACE}^{\text{RBSVs\_Jacquin's}}$ denote the ACEs of RBSVs for the proposed and Jacquin’s methods, respectively. NumRBSV denotes the number of range blocks with large variances. From Eqs.[24–26], we have

$$\text{Measure}=\text{Measure}^{\text{RBLVs}}+\text{Measure}^{\text{RBSVs}} \tag{27}$$

For example, for the boat image in Fig. 7(a), we have $\text{Measure}^{\text{RBLVs}}=-14{,}359.43 < 0$ which implies that for RBLVs, the proposed method can achieve smaller collage errors and provide a better block-matching effect. Although $\text{Measure}^{\text{RBSVs}}=13{,}751.68 > 0$, by Eq.[27], we have $\text{Measure}=-607.75<0$. Furthermore, by Eq.[24], we know that the proposed method can provide a smaller ACE, which implies a smaller ACER. Finally, by Eqs.[4, 5], we know that compared with Jacquin's method, the proposed method can provide better-decoded image quality.



(a) Boat (b) Peppers (c) Couple

(d) Airplane (e) Lake

Fig.8 Edge regions for the five test images.

Further, in the proposed method, SDBP is mainly designed to improve the quality of RBLVs, which are mainly located in the edge regions. Moreover, compared with the smooth regions, the eyes of people are more sensitive to the edge regions, and the quality of the edge regions plays a crucial role in influencing people's visual effects. Thus, we continue to compare the quality of the edge regions of the previous methods with that of the proposed method. Figure 8 shows the edge regions for the five test images, and white pixels represent the edges. We used $\mathrm{MSE}^{\mathrm{Edge}}$ in Eq. (28) to assess the quality of the edge regions in the decoded image regarding the counterpart in the original image, and a large $\mathrm{MSE}^{\mathrm{Edge}}$ implies poor image quality.

$$\mathrm{MSE}^{\mathrm{Edge}} = \frac{1}{\mathrm{NumE}} \sum_{i=1}^{\mathrm{NumE}} \left(\boldsymbol{p}_i^{\mathrm{Original\_Edge}} - \boldsymbol{p}_i^{\mathrm{Decoded\_Edge}} \right)^2 \qquad 28$$

where NumE denotes the number of pixels in the edge region. $\boldsymbol{p}^{\mathrm{Original_Edge}}$s and $\boldsymbol{p}^{\mathrm{Decoded_Edge}}$s denote the pixel values of the same edge region in the original and decoded images, respectively. Tab.2 implies that for the five test images, the corresponding average $\mathrm{MSE}^{\mathrm{Edge}}$ decreases from 211.92 to 195.47. We know that compared with Jacquin's method, the proposed method can provide smaller $\mathrm{MSE}^{\mathrm{Edge}}$, which imply better-decoded image quality in the edge regions.

From the fourth to seventh rows of Tab.1, we can see that compared with Jacquin's method, the average percentage of the required computations can be reduced from 100 to 79.72%, and the average bits per pixel of the proposed method can be reduced from 1.56 to 1.32. Thus, we know that the proposed method can provide fewer bits per pixel and reduce the computations required. For Jacquin's method, for an arbitrary range block, the corresponding block matching was restricted within conventional DBP. We had NumR=$(M/B)\times(N/B)$=4096 range blocks and NumD'=$(M/2B)\times(N/2B)\times8$=8192 domain blocks. In the encoding process, we should do block matching NumR×NumD'=33,554,432 times. For the proposed method, the block-matching operations of RBLVs were conducted within EDBP. Because the EDBP consists of DBP and SDBP, which contain NumS=(M/4B)×(N/4B)×8=2048 domain blocks, the EDBP contains NumED=NumD'+NumS=10240 domain blocks. For example, for the boat image in Tab.1, when BLAPACE reaches 95%, the NumRBLV=2728 range blocks are encoded, and we should do block matching NumRBLV×NumED=27,934,720 times. In the following encoding process, we adopted the no-search fractal encoding method for the remaining 1,368 RBSVs. Because the best-matched domain blocks of RBSVs are directly appointed, the corresponding block-matching operations can be saved, and the percentage of the computations required for the proposed method can be reduced to 27,934,720/33,554,432 = 83.25%. Contrarily, for each RBSV, we should only use 2 and 8 bits to quantize the scaling and offset coefficients, respectively. There is no need to store the isometric and position information of the best-matched domain blocks. Thus, the corresponding memory space can be saved. For example, for the boat image, the bits per pixel can be reduced from 1.56 to 1.35.

Moreover, from Eq. (23), the proposed method can adaptively divide the range blocks into two categories, RBLVs and RBSVs. For different images, the numbers of RBLVs that contribute to 95% of the total collage errors vary significantly, and the corresponding bits per pixel and computations required fluctuate significantly as well. For example, the percentage of the required computations decreased from Peppers' 90.97% to Airplane's 55.48%, and the bits per pixel decreased from Peppers' 1.42 to Airplane's 1.13.

Similar to Tables 1 and 2, Table 3 lists the performance comparison between Chaurasia's and the proposed methods. The average decoded image quality increased from 28.83 to 29.00 dB, average percentage of the required computations decreased from 100 to

64.89%, and average bits per pixel decreased from 1.56 to 1.33. Moreover, from Table 4, we observe that the average MSE^{Edge} decreases from 225.44 to 211.90. Clearly from Table 5, compared with Zheng's method, the average decoded image quality of the proposed method increased from 28.96 to 29.16 dB, average percentage of the required computations decreased from 100 to 70.90%, and average bits per pixel decreased from 1.56 to 1.37. From Table 6, the average MSE^{Edge} decreased from 222.54 to 206.13. From Table 7, compared with Gupta's method, the average decoded image quality of the proposed method increased from 29.03 to 29.19 dB, percentage of the required computations decreased from 100 to 64.19%, and average bits per pixel decreased from 1.56 to 1.33. From Table 8, the average MSE^{Edge} decreased from 217.98 to 203.57. In summary, compared with the previous methods, the proposed method can provide a better quality of decoded images, require fewer bits per pixel, and reduce required computations.

Table 1. Performance comparison between Jacquin's method and the proposed method.[1]

Test images		Boat	Peppers	Couple	Airplane	Lake	Average
PSNR(dB)	Jacquin's	28.28	31.80	28.47	29.46	27.72	29.15
	Jacquin's+Proposed	28.39	32.03	28.60	29.65	28.06	29.35
Computations needed(%)	Jacquin's	100%	100%	100%	100%	100%	100%
	Jacquin's+Proposed	83.25%	90.97%	89.54%	55.48%	79.35%	79.72%
Bits per pixel	Jacquin's	1.56	1.56	1.56	1.56	1.56	1.56
	Jacquin's+Proposed	1.35	1.42	1.40	1.13	1.32	1.32

Table 2. Comparison of MSE^{Edge}s between Jacquin's method and the proposed method.[1]

Test images		Boat	Peppers	Couple	Airplane	Lake	Average
MSE^{Edge}	Jacquin's	250.32	123.04	204.69	217.08	264.45	211.92
	Jacquin's+Proposed	235.47	111.97	191.17	203.41	235.31	195.47

Table 3. Performance comparison between Chaurasia's method and the proposed method.[5]

Test images		Boat	Peppers	Couple	Airplane	Lake	Average
PSNR(dB)	Chaurasia's	28.00	31.47	28.01	29.37	27.28	28.83

	Chaurasia's+Proposed	28.10	31.63	28.25	29.45	27.57	29.00
Computations needed(%)	Chaurasia's	100%	100%	100%	100%	100%	100%
	Chaurasia's+Proposed	68.36%	75.15%	72.12%	44.09%	64.75%	64.89%
Bits per pixel	Chaurasia's	1.56	1.56	1.56	1.56	1.56	1.56
	Chaurasia's+Proposed	1.37	1.43	1.41	1.13	1.33	1.33

Table 4. Comparison of MSE^{Edge}s between Chaurasia's method and the proposed method.[5]

Test images		Boat	Peppers	Couple	Airplane	Lake	Average
MSE^{Edge}	Chaurasia's	263.77	130.96	220.92	222.58	288.95	225.44
	Chaurasia's+Proposed	251.42	123.25	206.89	212.98	264.96	211.90

Table 5. Performance comparison between Zheng's method and the proposed method.[6]

Test images		Boat	Peppers	Couple	Airplane	Lake	Average
PSNR(dB)	Zheng's	28.04	31.78	28.00	29.29	27.70	28.96
	Zheng's+Proposed	28.15	31.99	28.15	29.49	28.04	29.16
Computations needed(%)	Zheng's	100%	100%	100%	100%	100%	100%
	Zheng's+Proposed	85.74%	76.22%	77.48%	47.59%	67.46%	70.90%
Bits per pixel	Zheng's	1.56	1.56	1.56	1.56	1.56	1.56
	Zheng's+Proposed	1.52	1.41	1.42	1.14	1.36	1.37

Table 6. Comparison of MSE^{Edge}s between Zheng's method and the proposed method.[6]

Test images		Boat	Peppers	Couple	Airplane	Lake	Average
MSE^{Edge}	Zheng's	265.78	123.69	230.40	226.85	266.00	222.54
	Zheng's+Proposed	251.20	113.24	216.04	212.31	237.84	206.13

Table 7. Performance comparison between Gupta's method and the proposed method.[7]

Test images		Boat	Peppers	Couple	Airplane	Lake	Average
PSNR(dB)	Gupta's	28.16	31.75	28.30	29.29	27.66	29.03
	Gupta's+Proposed	28.21	31.95	28.39	29.46	27.95	29.19
Computations needed(%)	Gupta's	100%	100%	100%	100%	100%	100%
	Gupta's+Proposed	67.38%	73.02%	73.00%	44.26%	63.28%	64.19%
Bits per pixel	Gupta's	1.56	1.56	1.56	1.56	1.56	1.56
	Gupta's+Proposed	1.36	1.42	1.42	1.13	1.32	1.33

Table 8. Comparison of MSE^{Edge}s between Gupta's method and the proposed method.[7]

Test images		Boat	Peppers	Couple	Airplane	Lake	Average
MSE^{Edge}	Gupta's	258.20	124.61	213.85	225.17	268.06	217.98
	Gupta's+Proposed	247.13	114.19	202.64	213.12	240.75	203.57

5 Conclusion

In this study, an adaptively hybrid fractal coding method was proposed to improve the performance of fractal coding methods. By analyzing the importance of the variances of range blocks, we found that RBLVs play a key role in determining the quality of decoded images. Thus, we designed EDBP for RBLVs to reduce the total collage errors. For the remaining RBSVs, the no-search fractal coding method, which can achieve fewer bits per pixel and computations required, was adopted. Moreover, if BLAPACE reaches 95%, an adaptive method for dividing the range blocks into two categories was proposed. In terms of the whole image, the proposed method improved the performance of fractal coding methods effectively and made them more useful in practical applications.

Acknowledgments

The work was partially supported by the State Scholarship Fund of China (Grant No. 201806575003). The authors would like to thank the reviewers for their time reading this paper.

References

1. A. E. Jacquin, “Image coding based on a fractal theory of iterated contractive image transformations,” *IEEE Trans. Image Process.* **1**(1), 18-30 (1992).
2. Y. Fisher, *Fractal Image Compression: Theory and Application,* Springer-Verlag (1994).
3. B. Wohlberg, and G. de Jager, “A review of the fractal image coding literature,” *IEEE Trans. Image Process.* **8**(12), 1716-1729 (1999).
4. A. E. Jacquin, “Fractal image coding: A review,” *Proc. IEEE.* **81**(10), 1451-1465 (1993).
5. V. Chaurasia, and V. Chaurasia, “Statistical feature extraction based technique for fast fractal

image compression," *Journal of Visual Communication and Image Representation* **41,** 87-95 (2016).

6. Y. P. Zheng, X. P. Li, and M. Sarem, "Fast fractal image compression algorithm using specific update search," *IET Image Process.* **14**(9), 1733-1739 (2020).
7. R. Gupta, D. Mehrotra, and R. K. Tyagi, "Hybrid edge-based fractal image encoding using K-NN search," *Multimed. Tools Appl.* **81**(15)**,** 21135-21154 (2022).
8. F. R. Shen, and H. Osamu, "A fast no search fractal image coding method," *Signal Processing: Image Communication* **19**(5), 393-404 (2004).
9. M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal image denoising," *IEEE Trans. Image Process.* **12**(12), 1560-1578 (2003).
10. M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal-wavelet image denoising revisited," *IEEE Trans. Image Process.* **15**(9), 2669-2675 (2006).
11. J. Lu et al., "An enhanced fractal image denoising algorithm," *Chaos Solitons & Fractals* **38**(4)**,** 1054-1064 (2008).
12. J. H. Jeng, C. C. Tseng, and J. G. Hsieh, "Study on huber fractal image compression," *IEEE Trans. Image Process.* **18**(5), 995-1003 (2009).
13. J. Lu, Z. X. Ye, and Y. Y. Zou, "Huber fractal image coding based on a fitting plane," *IEEE Trans. Image Process.* **22**(1), 134-145 (2013).
14. C. Xu et al., "A primal-dual algorithm for robust fractal image coding," *Fractals.* **27**(7), 1950119 (2019).
15. Y. Y. Zou et al., "A nonlocal low-rank regularization method for fractal image coding," *Fractals.* **29**(5), 2150125 (2021).

16. M. H. Pi, M. K. Mandal, and A. Basu, “Image retrieval based on histogram of fractal parameters,” *IEEE Trans. Multimedia.* **7**(4), 597-605 (2005).

17. X. Y. Wang, and Z. Chen, “A fast fractal coding in application of image retrieval,” *Fractals.* **17**(4), 441-450 (2009).

18. X. Huang, Q. Zhang, and W. Liu, “A new method for image retrieval based on analyzing fractal coding characters,” *Journal of Visual Communication and Image Representation* **24**(1), 42-47 (2013).

19. S. M. Abdullahi, H. X. Wang, and T. Li, “Fractal coding-based robust and alignment-free fingerprint image hashing,”, *IEEE Trans. Inform. Forensic Secur.* **15,** 2587-2601 (2020).

20. F. Khelaifi, and H. J. He, “Perceptual image hashing based on structural fractal features of image coding and ring partition,” *Multimed. Tools Appl.* **79**(27-28), 19025-19044 (2020).

21. Hong Pi, Hung Li, and Hua Li, “A novel fractal image watermarking,” *IEEE Trans. Multimedia.* **8**(3), 488-499 (2006).

22. F. Daraee, and S. Mozaffari, “Watermarking in binary document images using fractal codes,” *Pattern Recognit. Lett.* **35,** 120-129 (2014).

23. J. Lu et al., “A robust fractal color image watermarking algorithm,” *Math. Probl. Eng.* **2014,** 1-12 (2014).

24. King-Hong Chung, Yik-Hing Fung, and Yuk-Hee Chan, “Image enlargement using fractal,”, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing,* 273-275, VI-273 (2003).

25. C. M. Lai et al., “An efficient fractal based algorithm for image magnification,”, in *Proceedings of International Symposium on Intelligent Multimedia, Video and Speech Processing,* Hong Kong,

571-574, ii-iii (2004).

26. Z. P. Chen et al., “Image magnification based on similarity analogy,” *Chaos Solitons & Fractals* **40**(5)**,** 2370-2375 (2009).

27. Y. C. Wee, and H. J. Shin, “A novel fast fractal super resolution technique,” *IEEE Trans. Consumer Electron.* **56**(3), 1537-1541 (2010).

28. Z. Hua, H. C. Zhang, and J. J. Li, “Image super resolution using fractal coding and residual network,” *Complexity.* **2019,** 1-14 (2019).

29. S. Liu et al., “A fast fractal based compression for MRI images,” *IEEE Access.* **7,** 62412-62420 (2019).

30. Q. Wang, and S. Bi, “Prediction of the PSNR quality of decoded images in fractal image coding,” *Math. Probl. Eng.* **2016,** 1-13 (2016).